Preventing the Flood: Incentive-Based Collaborative Mitigation for DRDoS Attacks

Anat Bremler-Barr
Reichman University
Israel

Matan Sabag
Reichman University
Israel

Abstract*—Distributed denial of service (DDoS) attacks, especially distributed reflection denial of service attacks (DRDoS), have increased dramatically in frequency and volume in recent years. Such attacks are possible due to the attacker's ability to spoof the source address of IP packets. Since the early days of the internet, authenticating the IP source address has remained unresolved in the real world. Although there are many methods available to eliminate source spoofing, they are not widely used, primarily due to a lack of economic incentives. We propose a collaborative on-demand route-based defense technique (CORB) to offer efficient DDoS mitigation as a paid-for-service, and efficiently assuage reflector attacks before they reach the reflectors and flood the victim. The technique uses scrubbing facilities located across the internet at internet service providers (ISPs) and internet exchange points (IXPs). By transmitting a small amount of data based on border gateway protocol (BGP) information from the victim to the scrubbing facilities, we can filter out the attack without any false-positive cases. For example, the data can be sent using DOTS, a new signaling DDoS protocol that was standardized by the IETF. CORB filters the attack before it is amplified by the reflector, thereby reducing the overall cost of the attack. This provides a win-win financial situation for the victim and the scrubbing facilities that provide the service. We demonstrate the value of CORB by simulating a Memcached DRDoS attack using real-life data. Our evaluation found that deploying CORB on scrubbing facilities at approximately 40 autonomous systems blocks 90% of the attack and can reduce the mitigation cost by 85%.

I. Introduction

Today, almost twenty years after the first large-scale distributed denial of service (DDoS) attack, DDoS attacks still constitute one of the biggest threats to the legitimate use of the internet [1]. The number of DDoS attacks continues to increase due to the proliferation of Internet of Things (IoT) devices and cloud services that are exploited for massive malicious botnets. There is a disturbing economic system behind the flourishing DDoS attacks. It costs less than 100$ to launch a 2Tbps attack on the DDoS market, based on botnets as a service. An attacker can make thousands of dollars in extortion this way, creating a clear arbitrage [2]. The potential victims, which can be any network on the Internet, end up having to pay a relatively high monthly fee of 5000$ or more to hire defense facilities for DDoS mitigation [2]. These scrubbing facilities can be installed either on-premises, at the internet service providers (ISPs), at internet exchange points (IXPs), or in the cloud (e.g., Prolexic, Akamai, Cloudflare, and others). To cope with the increasing size of DDoS attacks, these defense facilities race to acquire more computing power and bandwidth that typically remains idle waiting for a large attack to occur.

This problem has motivated a number of suggestions for collaborative defense schemes [1], [3]–[8]. The idea is to eliminate the imbalance and wasted resources by having different scrubbing centers or even router capabilities working in collaboration across networks to mitigate a single attack. We are also seeing the standardization of signaling information about these attacks, such as the IETF DDoS Open Threat Signaling (DOTS) standard [9]. This information is intended to provide real-time signaling of DDoS-related telemetry and DDoS handling requests to assist collaboration in detecting, classifying, and mitigating DDoS attacks.

The largest attacks observed are distributed reflection attacks (DRDoS). These attacks have grown in size from 300 Gbps in 2013 to 3.47 Tbps in 2022 [10]–[20], as seen in Figure 1. Distributed reflection denial of service (DRDoS) attack is a specific type of DDoS. In a DRDoS attack [21], [22], the attacker sends query packets to reflector servers using an IP address spoofed to the victim's, overwhelming them with response packets. Reflectors are open servers that are usually UDP-based services, such as DNS resolution, memcached servers, or NTP. Reflection attacks are prevalent because attackers can send surged queries to the reflectors that trigger larger responses by order of magnitude, up to 50,000 [23]. Moreover, in reflector attacks, the attacker's identity is hidden from the victim since the traffic comes from legitimate servers.

An attacker might also use carpet bombing [24], [25], a variation of DRDoS. Instead of targeting a single destination, the attacker attacks many addresses within a specific subnet to choke the network.

DRDoS attacks are only feasible due to the ability to spoof an IP address. In this context, authenticating the IP source address remains one of the most pressing issues in combatting internet attacks. The research community has proposed a variety of solutions [3], [26]–[36], but almost none have been widely adopted because there is no financial incentive for deployment. By using these mechanisms, a network does not protect itself from being targeted by a spoofed attack. Instead, it acts altruistically by preventing the source from launching an attack and helping other networks under attack.

ISBN 978-3-903176-48-5© 2022 IFIP

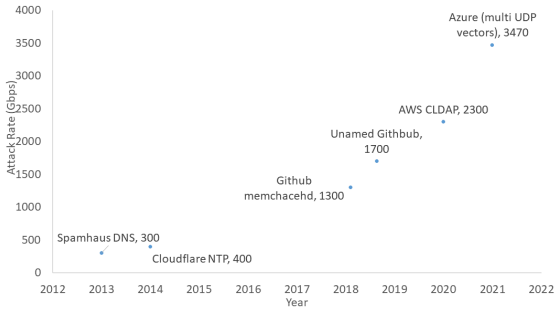

Fig. 1. Growth in volume of DRDoS attacks

This paper proposes a Collaborative On-Demand Route Based (CORB) DRDoS mitigation defense scheme to combat reflector attacks by using BGP route information. CORB is based on several simple observations:

- **Collaborative defense is necessary:** To cost-effectively deal with the increasing number of DDoS attacks over the global internet, it is essential to collaborate and facilitate the power to mitigate across organizations. Financial incentives are essential to motivate collaboration, and smart-contracts such as those used in blockchain technologies can make payment for collaborative defense more feasible [5].
- **Attack mitigation is best performed closer to the source, before amplification takes place:** Before amplification, the volume of traffic that needs to be handled is much smaller. In some cases, using a solution closer to the victim can be too late because the network has already been saturated [1]. Mitigation costs are also higher once the attack reaches the victim because they are, in most cases, proportional to traffic volumes. Moreover, in DRDoS, it is more difficult to identify the attacked packets after amplification. This is because they are not spoofed since they are triggered from the reflector to the victim.
- **Route based mitigation is practical using BGP information:** In DRDoS, packets are spoofed to appear to originate from the victim, but their actual routing differs from the route taken by a legitimate packet sent by the victim. These cases can be identified by examining the victim's BGP information, which reveals the route information. Because many attacks today are multi-vector attacks, using multiple reflectors and protocols, one of the advantages of CORB is that it addresses the root cause to identify spoofing regardless of the protocol. We explain this simple idea and discuss different approaches to deploying CORB in Section III. Previous theoretical papers have already highlighted BGP information as a valuable resource. However, they made some unrealistic assumptions, such as changing the BGP protocol (see Section VI). We present a feasible and practical solution. In CORB's basic mode (CORB-loose), the victim's signal to the scrubbing facilities is merely a list of headers to filter, where the source is the victim, and the destination is the reflector. Based on the information derived from the attack traffic and the victim's BGP table, the information is signaled in real time (e.g., through the DOTS protocol). The technique has zero-false positives and it does not assume anything about the characteristics of the BGP routes, such as whether or not they are symmetric.
- **Mitigation based on the source address:** Mitigation solutions are usually tailored for traffic in which the destination address is the target of an attack. In order to effectively mitigate an attack before reflection occurs, mitigation solutions should target traffic whose source address matches that of the victim. Note that this is legally acceptable, since the victim is–or appears to be–the owner of packets. Therefore, the mechanism is also GDPR compliant from a privacy perspective. As far as authentication is concerned, the same mechanism should be applied to mitigation by source address as it is to mitigation by destination address. Moreover, current standard mechanisms such as BGP Flowspec [37], allow filtering based on information such as the source address.

The remainder of the paper is organized as follows. Section II provides an overview of the CORB architecture at a high level. Section III examines CORB's implementation details in detail, demonstrating that it is both feasible and practical. By simulating the Github Memcached attack in 2018, Section IV illustrates how CORB can mitigate DRDoS attacks before amplification occurs, drastically reducing network bandwidth consumption and mitigation costs. Section V examines implementation details pertaining to DOTS. Chapter VI covers related work.

Although this work focuses on DRDoS attacks, our mechanism can be applied with other collaborative defense techniques. As the size of the attack increases, we believe it becomes essential to implement a collaborative solution that offers financial incentive to mitigate DDoS attacks and deal with this growing threat. This paper is a step towards that goal.

II. CORB Architecture

CORB is designed to subdue large scale reflection attacks by combining the mitigation and scrubbing power of different facilities spread across the globe. The algorithm uses BGP information from the victim to identify the attack traffic. BGP is a standardized protocol used to exchange routing and reachability information between autonomous systems (AS). Each AS is a collection of networks that are defined by prefixes and are under the control of a single administrative operator (i.e., internet service provider or a large enterprise). CORB takes advantage of the scrubbing capabilities of AS networks spread across the internet, and aggregates their power to mitigate DDoS attacks.

DDoS mitigation services are available in a variety of forms: in ISPs, in IXPs, and as a service in the cloud. We now examine how CORB can take advantage of these services,

and highlight the technologies that can be used to implement CORB, such as BGP Flowspec, and DOTS.

- Scrubbing at the ISP level: ISPs can deploy dedicated scrubbing devices (e.g., Radware), but we demonstrate that CORB can be implemented using standard protocols for DDoS mitigation. The basic and most common technique used by ISPs on their border routers is Remote Triggered Blackhole (RTBH) [38], [39]. Based on its source or destination address, RTBH is able to abandon undesirable traffic before it enters the AS, by dropping traffic in all the BGP border routers. BGP Flowspec [37] is a more sophisticated and subtler way to block specific flows of traffic based on their destination, source, port, DSCP, and others. BGP Flowspec is the control protocol that injects filters to the data plane of the AS border routers (ASBRs), which can handle filtering at a high rate. Because BGP Flowspec can apply filtering rules to all of the AS's border routers, we refer to the AS as a single entity throughout the remainder of this paper.
- Scrubbing at the IXP level: Internet exchange points (IXPs) provide DDoS mitigation services similar to those provided by ISPs [40]. IXPs offer an ideal location at which to deploy DDoS mitigation, since many ISPs make use of them for traffic exchange. For example, the DE-CIX in Frankfurt and the AMS-IX in Amsterdam, use over 800 networks and over 6 Tbps [41]. By empowering one of these large IXPs with mitigation services, hundreds of member networks will immediately benefit, without the need for changes in Internet protocols or cooperation and coordination between the member networks. We demonstrate that applying CORB to large IXPs can significantly enhance attack mitigation.
- Scrubbing with cloud scrubbing centers: Cloud-based scrubbing centers are another form of DDoS mitigation services. In the event of an attack, traffic is redirected via DNS or BGP to a scrubbing center, where a mitigation system subdues the attack traffic and forwards clean traffic back to the network for delivery. The majority of scrubbing vendors [42] have multiple scrubbing centers that can handle large-scale attacks situated at different locations around the globe [43], [44]. It is noteworthy that the attack volume of the most recent DRDoS attack is close to the maximum capacity of many of these scrubbing centers [44]. CORB assumes that we are aware of the legitimate packet route and, as a result, all other routes are deemed illegitimate; however, BGP and DNS redirection can cause the traffic to change routes, making it more difficult to maintain the original route information. Hence, this work focuses on the use of mitigation services for DDoS within the AS network, primarily at ISPs and IXPs.

CORB requires that we transmit information related to the attack from the victim to the scrubbing ASes (SASes). To accomplish this task, we use DDoS Open Threat Signaling (DOTS) [9], [45]. DOTS is a protocol that was standardized by the Internet Engineering Task Force (IETF) and is designed for real-time signaling related to DDoS attacks.

Lastly, it is important to note that the CORB algorithm does not assume any properties for the internet routes; they can be asymmetric, and the victim may be an anycast address.

III. CORB

CORB is based on the principle that even if an attacker succeeds in spoofing packets that bear the victim's source address, those packets will have been sent to the reflector via routes that are different from those typically used for legitimate packets originating from the victim. The expected route from the victim to the reflector can be determined at the AS level by using BGP information. Each destination (i.e., a network prefix) in the BGP table is assigned with an outgoing neighbor and an associated ASpath. BGP is a policy routing protocol that chooses the best route according the ASpath, which is the route expressed in AS level information. As part of CORB, we use the ASpath to extract the pertinent filtering information and send it to the scrubbing ASes (SASs). By examining the ASpath information, the scrubbing AS can identify which packets did not arrive via the expected route. With this information in hand, the scrubbing AS can filter out a majority of the attack volume prior to reflection and amplification. [2]

In this section, we have a victim v and M scrubbing ASes $sAS_1, ..sAS_M$ spread around the world. There are two modes of CORB operation: **CORB-Strict** and **CORB-Loose**, where loose transmits less filtering information and can filter fewer cases, as compared with strict. The algorithm consists of three stages: detection, signaling, and mitigation.

A. Detection

The paper assumes that the victim v has a basic DDoS detection mechanism capable of identifying DDoS attacks if a substantial amount of packets are involved. In addition, we assume the detection mechanism can classify the attack type as a DRDoS attack and discover the set of reflectors $R = r_1, r_2, ...r_k$ involved in the attack based on their IP addresses. Detecting a DRDoS attack and determining the reflectors involved is typically straightforward. This is because the victim will notice that it receives responses even though it did not initiate the request [46]–[49].

B. Signaling

Upon detecting the DDoS attack, the victim sends the scrubbing ASes signals requesting mitigation. The mitigation request contains information retrieved from the victim's BGP table. The specifics of what is sent depend on the CORB mode. The signaling can be performed using DOTS [9].

Figure 2 illustrates the basic concept behind **CORB-Loose**. In this scenario, the expected route of an authentic packet sent

[2]If the victim is a server, the BGP information is held by the server's internet service provider (ISP). This is good news because ISPs are usually involved in the DDoS mitigation process to prevent attacks from disrupting the entire network.

from the victim v to the reflector r is not expected to pass through the scrubbing AS sAS. In other words, the scrubbing ASes should not observe packets from v to r unless they are spoofed. For this reason, the scrubbing AS can filter all of these packets before reflection, thereby reducing the flood that would have been caused by an amplification.

For each reflector r participating in the attack and a scrubbing AS sAS, the victim v performs the following. It checks the BGP table to see whether the BGP route (ASpath) from itself to r traverses through sAS. When it does not, v sends a mitigation request to sAS to filter all packets from v to r. [3]

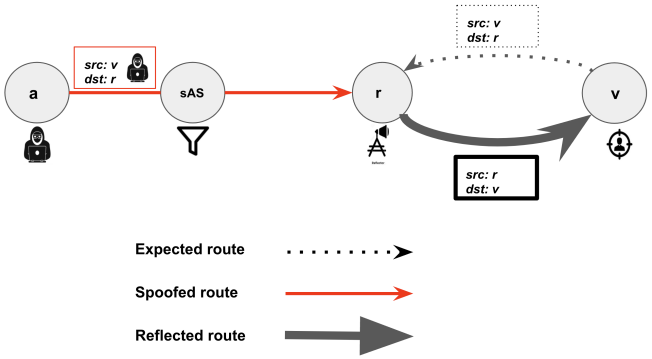

Fig. 2. CORB-Loose: sAS filters spoofed traffic arriving from the attacker a with destination r and source v because sAS does not reside on the expected route between v to r

CORB-Strict deals with scenarios where sAS is on the expected route from the victim to r, which prevents CORB-Loose from detecting and mitigating it. But in this case, the incoming interface is different, as shown in Figure 3. Here, the victim v sends to sAS information about the expected incoming interface a legitimate packet from itself to r should enter to sAS. The incoming interface information is passed in the form of the neighbouring AS Number (ASN). If sAS is not included in the path to r, it sends $null$. In our example, this information is the neighboring AS n. Hence, sAS should filter out any packets going from v to r that do not come through the neighboring AS n. Consequently, CORB-Strict can filter a larger number of spoofing cases.

Providing the BGP information is current, it is straightforward to prove that the algorithm has zero false-positives. In addition, the algorithm does not assume any properties for the internet routes; they can be asymmetric, and the victim may be an anycast address. Hence, we should ensure that the signaled information is current, i.e., we should notify the scrubbing AS if the information provided to it has changed due to changes in BGP routes. The updates should be minimal since most BGP routes on the internet remain stable for days [50] and the likelihood of any significant change in relevant information is small. If there are new reflectors involved in the attack, this would be another reason for updating to occur.

[3]It is imperative to note that the lookup in the BGP table is based on the IP address of the reflector and not on its AS. This is because some subnets within the same AS may have different routing.

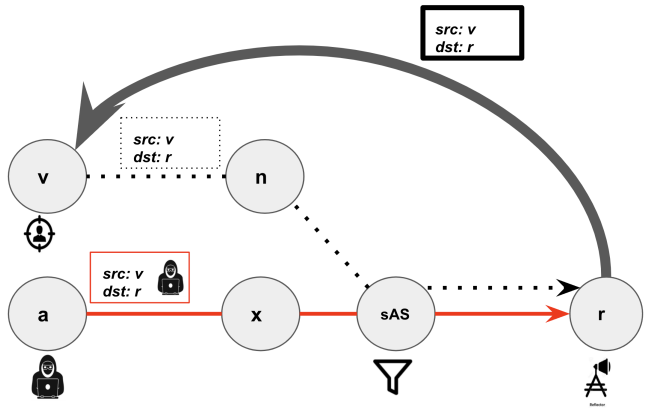

Fig. 3. CORB-strict: sAS filters spoofed traffic arriving from the attacker a with destination r and source v because it arrives from the incoming interface x while legitimate traffic from v to r should enter through the incoming interface n. Note that CORB-Loose cannot detect this situation because sAS is indeed on the path from v to r.

Signaling to a scrubbing AS in CORB-Loose has a size bound of $O(|R|*32)$ for IPv4 and $O(|R|*128)$ for IPv6, where $|R|$ is the number of reflectors. In reality, this size bound will be significantly lower than $|R|$ since we send the IP address of a reflector only when the observed traffic from the victim must be filtered by the scrubbing AS. In CORB-Strict, the bounds are $O(|R| * (32 + 32)$ and $O(|R| * (128 + 32)$, respectively, since for each reflector we also include the neighbor ASN of the scrubbing AS that is expected to send the route legitimate traffic from the victim to sAS. We note that the memory state is proportional to the number of reflectors, but not to the number of attackers, which is significantly higher.

C. Mitigation

In order to detect spoofed traffic, packets bearing the source address of the victim should be filtered in accordance with CORB filtering information. As noted previously, the Flowspec protocol supports filtering based on both the source and destination addresses. **CORB-Loose** filtering is straightforward, as all traffic claiming to originate from the victim must be dropped. **CORB-Strict** requires filtering according to the information pertaining to the incoming interface from which it entered the AS. This can be achieved with BGP Flowspec and VRF (Virtual routing and forwarding) features.

D. IXP case

So far, we have discussed the capabilities of performing scrubbing in ASes (scrubbing ASes). An additional candidate for scrubbing is the Internet Exchange Point (IXP), where many Internet infrastructure companies, such as Internet Service Providers (ISPs), exchange Internet traffic. Recent studies suggest that many IXPs also suggest mitigation and detection techniques [40], [41]. We can therefore see that the IXP provides a filtering facility that works on the edges between two ASes. The CORB algorithm can be adapted naturally to IXP, checking edges instead of AS. As an example with

CORB-Loose, Let (AS_s,AS_e) be a link in an IXP that has scrubbing capabilities. For each reflector r participating in the attack and a link (AS_s,AS_e), the victim v performs the following. It examines the BGP table to determine whether the BGP route (ASpath) from itself to r traverses the link (AS_s,AS_e). When it does not, v sends a mitigation request to the corresponding IXP to filter all packets from v to r.

E. Carpet mitigation attack

It is straightforward to implement CORB to mitigate carpet bombing attacks [24], [25]. This is a variation of DRDoS that, instead of attacking one destination, attacks a whole subnet of addresses to choke a network. In this case, CORB should be applied to the victim subnet.

F. Accuracy

CORB has almost no false-positives since false-positives occur only during the updating of the BGP information. That said, under DDoS attacks, some false positives are reasonable since the victim would not be available regardless of mitigation. Inherently, CORB has some false-negative states. This means it can reduce the volume of the attack but not eliminate it completely, which is common to many mitigation DDoS solutions [28]. This is because most mitigation techniques are used in layers, where each mitigation technique filters some of the traffic until it achieves a reasonable volume that the victim can handle. An obvious false-positive case occurs when there is no scrubbing center between the attacker and the reflector; in other words, when CORB is not in the spoofed traffic path. To analyze the limitations of CORB (false positives), we define $LCA(a, v, r)$ as the least common ancestor at the AS level between the legitimate route from v to r and the spoofed route from attacker a to r. As long as sAS appears before $LCA(a, v, r)$, both CORB modes are useful because sAS does not expect to see traffic from v (see Figure 2). When sAS appears after $LCA(a, v, r)$, both modes result in false-negatives since the incoming interface is the same for both legitimate and spoofed traffic, as seen in Figure 4. It is only in the case of $LCA(a, v, r) = sAS$ that CORB-Loose may produce false negatives, but this is not the case for CORB-Strict (see Figure 3). Accordingly, the difference in performance between the two modes is not that significant, as we demonstrate in Section IV.

IV. CORB Simulation

The primary objective of CORB is to protect against spoofed reflected traffic. We first define measures of filtering performance that apply to both strict and loose modes. We measure impact of scrubbing in the AS level. an AS that deployed CORB is denoted as a scrubbing AS (SAS).

A packet sent from AS a to AS r that spoofs AS v will be filtered out by SAS sAS if and only if the packet's route includes sAS and if sAS is capable of filtering out the spoofing of v under CORB version ver, which can either be loose or strict. We represent the aforementioned statement as follows:

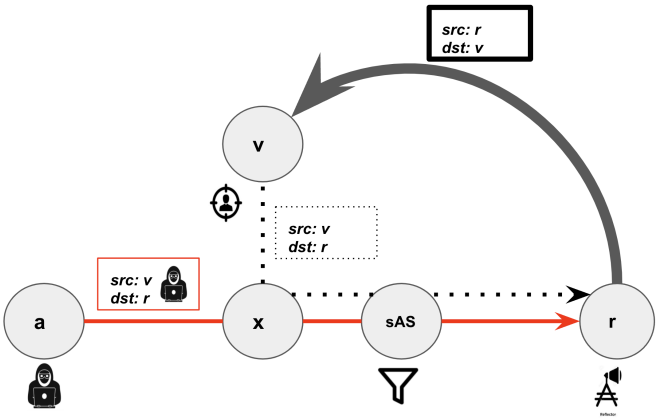

Fig. 4. Both spoofed traffic from the attacker a to r and legitimate traffic from the victim v to r are routed through x before arriving at the scrubbing center sAS. Therefore, both legitimate and spoofed traffic arrive at sAS on the same interface, and both CORB modes will fail to detect spoofing.

$$filter^{ver}_{sAS}(a, r, v) = \begin{cases} 1 & sAS \text{ filtered the packet} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Similarly, we can define the $filter$ function for a set of SASes S.

$$filter^{ver}_{S}(a, r, v) = \begin{cases} 1 & \exists\ sAS \in S \text{ which filtered the packet} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

DDoS defense must be examined from the perspective of the victim, taking into account the attackers, reflectors, and scrubbing ASes. Let S represent the set of all SASes, R the set of all reflectos, V the set of all victim ASes, and G the set of all ASes in the graph. **Cumulative reflection protection** (CRP) is the number of paths in the graph in which packets sent from any AS $a \in G$ to any AS $r \in R$, spoofing an address from AS $v \in V$, are filtered by a $sAS \in S$.

$$CRP^{ver}(V, S) = \sum_{a \in G} \sum_{r \in R} \sum_{v \in V} filter^{ver}_{S}(a, r, v) \quad (3)$$

This measure contains the possible paths by which scrubbing AS $sAS \in S$ can filter traffic from an attacker AS $a \in G$ to a reflector AS $r \in R$, which spoofs victim AS $v \in V$. Essentially, this measurement describes the filtering ability of CORB in the context of a specific attack. We define the percentage of successful mitigation (% mitigation) as the ratio between $CRP(V, S)$ and the total number of possible reflection attack paths. Formally:

$$\%mitigation = \frac{CRP(V, S)}{\sum_{a \in G} \sum_{r \in R} \sum_{v \in V} 1} \quad (4)$$

Therefore, in this section, when we say, for example, that a configuration of CORB **blocks** 34% percent of the attack, it

is meant that the ratio between the $CRP(V, S)$ and the size of all possible attack vectors in the simulation is 0.34.

A. Memcached Attack Simulation

We drew inspiration from the 1.3 TBps memcached attack on GitHub in 2018 to conduct a realistic simulation [23]. The AS graph was built based on Caida's AS Relationship data for 2019-08-01 [51]. The path between pairs of ASes was calculated using the $bgp - simulator$ tool [52] that we developed, which was heavily based upon the disco tool [53] that infers the BGP routing based upon [54]. Nevertheless, this was done only for the simulation, and CORB does not assume that paths between ASes are determined in this manner. Using shodan data [55], we defined the set of reflector ASes R as those with more than ten open memcached IP addresses. V is defined as the set of victim ASes. At the time of the incident, V contained two ASes that GitHub resided on. All ASes are potential attackers in this simulation. The only difference between simulations is the SAS set S set and the CORB mode. Within the following sections, we elaborate on how we selected S in order to maximize CORB's efficiency. We run the experiment twice for each choice of S, once with CORB-strict (Fig 5) and once with CORB-loose (Fig 6).

In this simulation, we do not account for updates of BGP routes in the graph during the attack; this does not affect the algorithm itself and is only done for convenience. Because CAIDA's data is incomplete, we avoided scenarios in which the path between two ASes could not be calculated. Fortunately, this is a small proportion (less than 1%) of the paths in the graph.

B. Simulation results

In this section, we present and analyze our simulation results that are reflected in Fig. 5 and in Fig. 6 for strict and loose CORB modes respectively. We show that in order to achieve route-based packet filtering efficiently with a relatively small number (fewer than 100 out of 65,000) of SASss, we are required to choose core networks that tend to be the LCA of many ASs.

- **Tier-1**: When S is set to tier-1 ASes, CORB-Strict and CORB-Loose can respectively block 44% and 41% of attacks. Furthermore, CORB-Strict and CORB-loose block 40% and 37% of the attack respectively, just by deploying on ten tier-1 ASes. If we assume that a small number of ASes will implement a sophisticated defense system for economic reasons, we will witness an immediate reduction in the volume of attacks on the internet. In selecting only tier-1 ASes, one can expect that CORB will have no effect when an attack targets further downstream tiers, as no SAS is present in any of the attack paths.
- **Tier-2**: CORB-Loose and CORB-Strict block 37% and 22% of attacks, respectively, when deployed on tier-2 ASes. According to our estimate, this is drastically less efficient than deploying on tier-1 ASes, since tier-2 ASes are LCAs of fewer ASes.
- **Tier-1 and tier-2**: Using both tier-1 and tier-2 ASes, we created a combined list, and we selected the top 100 with the most direct customers. There was a dramatic improvement, as five SASs were able to block more than 50% of the attack, and 25 SASs were able to block 70%.
- **By-customers**: S is defined as the top 100 ASes according to the number of direct customers. In choosing based upon direct customers, it implies that there are a large number of ASes whose LCA is in S. This method of deployment has been found to be the most effective. This deployment is less effective than the Tier-1 and Tier-2 deployments for the first 18 SASs. However, while the latter keeps stagnating after reaching approximately 70%, this deployment reaches 92% protection with 84 filtering SASs (with CORB-Strict), more than 20% higher than deployments using tier-1 or tier-2 ASes. Furthermore, we observe that 80% mitigation can be achieved by 70 SASs, rendering the attack practically insignificant. CORB-Loose offers protection up to 80% and corresponds to a performance degradation of approximately 12% compared to strict CORB. Yet, CORB-loose in this scenario is still superior to the other deployments mentioned, even if in strict mode. CORB will therefore have an influential impact on attack mitigation when deployed on services such as IXPs. They serve as the link between internet infrastructure companies such as ISPs and CDNs, thereby serving as the LCA for a number of networks.
- **Random** We carried out 20 iterations of random deployments of 100 SASs. Random deployments demonstrated poor performance when using route-based packet filtering. The best results came from random deployments that were less than 2%. Stub ASes and ASes that serve as the LCA for a small number of ASes have no effect and are ineffective. This illustrates the importance of selecting SASs carefully. Fortunately, the ASes that are suitable for CORB deployment are also the ones with an incentive to do so.

C. Financial implications

By dropping spoofed packets before reflection, CORB reduces attack volume significantly. It is reasonable to assume that the Scrubbing ASes will charge a fee based on the volume of traffic. The ratio between the sizes of the response and the request is called amplification factor. For amplification factor P, handling a spoofed packet after reflection is more expansive by approximately P. Accordingly, CORB has a profound effect on mitigation costs for victims.

Memcached has a bandwidth amplification factor of about 50,000 [23]. To estimate the victim's saving when using CORB, we charged 1 money unit (such as a dollar) for packets that CORB was able to filter, and 50,000 for packets that CORB was not able to filter. This price was compared to the cost it would have paid without CORB deployment, i.e. all spoofed packets reach their targets.

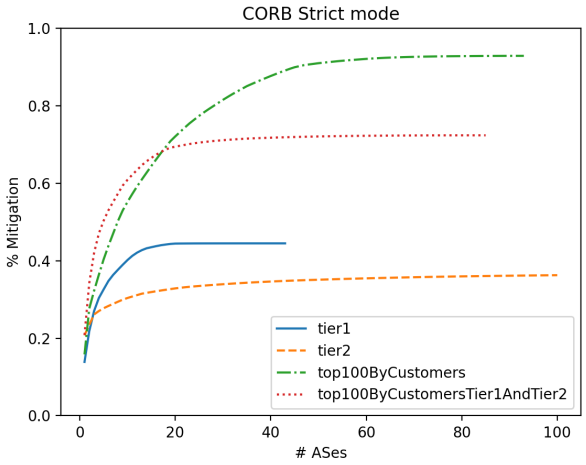

Fig. 5. Attack simulation with strict CORB

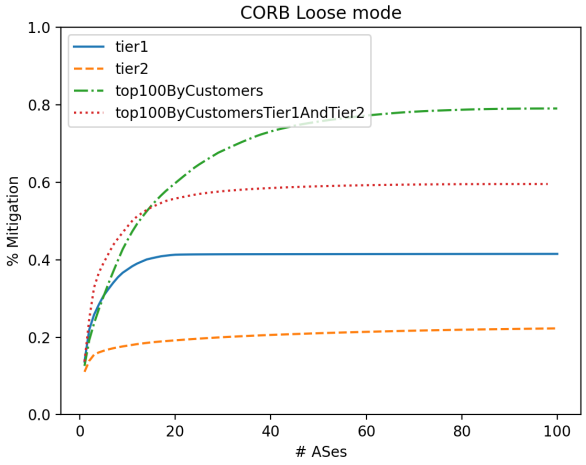

Fig. 6. Attack simulation with loose CORB

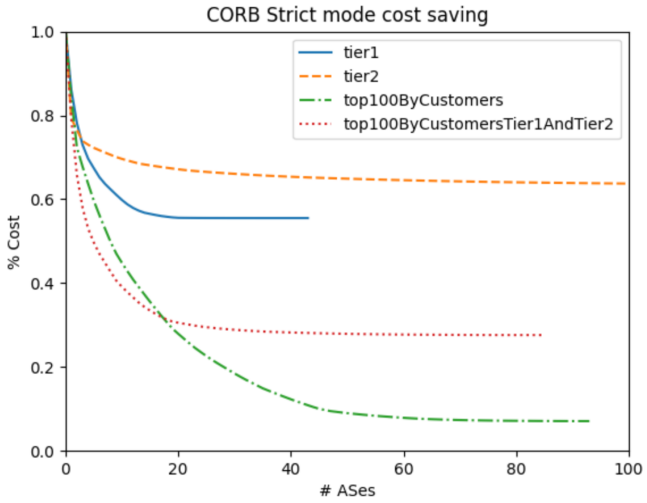

Fig. 7.

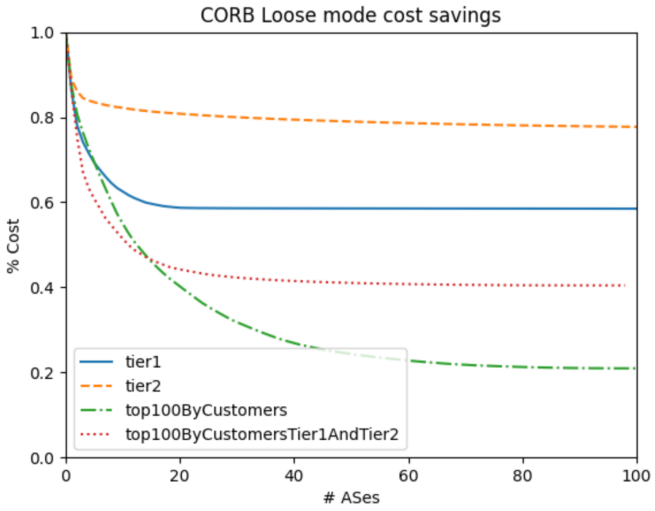

Fig. 8.

Figure 7 illustrates CORB-Strict's ability to save from 25 percent when deployed on the least effective SAS set (tier-2) to 90 percent when deployed on the most effective SAS set. The graph in figure 7 clearly depicts the opposite of the graph in figure 5 due to the relationship between mitigation success rate and cost savings. In addition, CORB can save us approximately 22% to 65%, based on the SAS set, even if we only use it on 20 SASs. CORB-Loose cost savings are slightly less than CORB-strict ones, but still of paramount importance to the victim. By deploying CORB-Loose, victims could save between 10 and 80 percent, depending on the SAS set (2).

It is evident from these results that both the SASs and the victims can benefit from the deployment CORB.

V. DOTS Implementation Details

This section describes how DOTS should be extended to support CORB. Our offer to extend DOTS, as opposed to, for example, designing our proprietary protocol, is based on the fact that in order to be widely adopted, a defense mechanism must be simple to implement and based on standard internet protocols.

There are two types of protocols between a DOTS client and a DOTS server: the signal channel [45] and the data channel [56]. DOTS clients use the signal channel in order to signal their need for DDoS protection. If the DOTS client detects a DDoS attack, it sends a mitigation request over the signal channel, which includes the following fields: target-prefix, target-protocol, etc. It can be concluded from this that DOTS is primarily concerned with mitigating DDoS attacks based on the target, which requires examination of the packet's destination field.

As mentioned in section III, the victim of a reflection attack may be interested in requesting spoofed traffic filtering based on the packet's source address. In CORB-Loose, a request by the victim to the SAS to filter packets based on the

victim's source address is accompanied by a list of reflectors to which packets from the victim are not expected to be forwarded through the SAS. When using CORB-Strict, the additional information being sent is a list of the reflectors participating in the attack and their expected incoming interface when authentic packets are routed through the SAS (or null, if there is none). In order to send such information with DOTS, we offer to extend the signal channel protocol by adding a $target_reflector_interface$ field entry to the mitigation request. As this is an optional entry, backward compatibility can be assured (earlier versions of DOTS will ignore it). The $target_reflector_interface$ contains a list of ⟨reflector_network,asn⟩ tuples. Each tuple contains the IP address of the reflector and its corresponding expected incoming interface, which is always $null$ in CORB-Loose. Unlike in our simulation, in which we addressed the reflector's AS as a single unit, we specify the reflector's address because in reality, different networks within the same AS may have different routes to the same destination.

VI. Related Work

DDoS mitigation and IP spoofing prevention are crucial for a functioning and stable internet environment. As such, the research community has spent years proposing solutions to these topics.

The majority of these solutions, such as [26], [27], [29]–[33], [36], suffer from one or more of the following: they do not provide incentives for deployment, they are not feasible in today's internet environment, or they make assumptions that are unrealistic.

In the past years, many papers proposed solutions that provide some form of cooperative defense. Some solutions such as [4], [57] are unable to protect against attacks from legacy networks without their defenses in place. Others, such as [58] do not perform well in non-contiguous deployment. Some papers are tailored to a small group of ISPs that are interconnecting with one another and have the relationship of customer-provider or peer, a limitation that CORB is not subject to. [59]–[61] have developed collaborative approaches for detecting and neutralizing botnets that participate in attacks and for developing collaborative IP blacklists and collaborative IP blacklists. These may suffer from deficiencies as they are not well maintained and properly updated.

CORB resembles approaches like Catchit [3] in that it is a collaborative route-based solution that provides routers with information regarding the valid arrival path of packets from a specific source. CatchIt has several shortcomings, such as the requirement to check every packet, even if no attack is in progress, and the necessity to continuously update a table that maps between sources and corresponding interfaces. Perhaps the most significant difference between CORB and CatchIt is that CatchIt both modifies BGP as well as introduces a proprietary protocol called RING to establish the route-sharing mechanism between the ASes. As a result, ASBR's control and data planes will need to be modified. Taking advantage of standard protocols such as DOTS and BGPFlowspec, CORB overcomes this problem.

VII. Conclusion

CORB is a novel route-based filtering system for mitigating DRDoS attacks that employs network collaboration in order to hamper even the most powerful attacks. The most important contribution of our work is that we establish a practical framework that enables routers to receive information about inter-domain BGP routing decisions on-demand by utilizing existing protocols while providing true economic incentives for networks to adopt it.

References

[1] E. Osterweil, A. Stavrou, and L. Zhang, "21 years of distributed denial-of-service: A call to action," *Computer*, vol. 53, pp. 94–99, 08 2020.

[2] C. Labovitz, "Tracing volumetric ddos to its booter / iphm origins," 2021. [Online]. Available: https://www.nanog.org/news-stories/nanog-tv/nanog-82-webcast/tracing-ddos-end-to-end-in-2021/

[3] J. Li, J. Bi, and J. Wu, "Towards a cooperative mechanism based distributed source address filtering," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, 2013, pp. 1–7.

[4] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "Cossack: Coordinated suppression of simultaneous attacks," in *Proceedings DARPA Information Survivability Conference and Exposition*, vol. 2, 2003, pp. 94–96 vol.2.

[5] M. Essaid, D. Kim, S. H. Maeng, S. Park, and H. T. Ju, "A collaborative ddos mitigation solution based on ethereum smart contract and rnn-lstm," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–6.

[6] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, "A framework for a collaborative ddos defense," in *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, 2006, pp. 33–42.

[7] R. Saad, F. Nait-Abdesselam, and A. Serhrouchni, "A collaborative peer-to-peer architecture to defend against ddos attacks," in *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, 2008, pp. 427–434.

[8] B. Rashidi, C. Fung, and E. Bertino, "A collaborative ddos defence framework using network function virtualization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2483–2497, 2017.

[9] R. Moskowitz, "Ddos open threat signaling (dots) requirements," 2019.

[10] "Network infrastructure security report," 2011. [Online]. Available: http://www.arbornetworks.com/report

[11] M. Prince, "The ddos that knocked spamhaus offline," 2013. [Online]. Available: https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho

[12] ——, "Technical details behind a 400gbps ntp amplification ddos attack," 2014. [Online]. Available: https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack

[13] "Memcached-fueled 1.3 tbps attacks," 2018. [Online]. Available: https://blogs.akamai.com/2018/03/memcached-fueled-13-tbps-attacks.html

[14] "Memcached ddos attack." [Online]. Available: https://www.cloudflare.com/learning/ddos/memcached-ddos-attack/

[15] "Inside the infamous mirai iot botnet: A retrospective analysis," 2017. [Online]. Available: https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/

[16] T. Bienkowski, "No sooner did the ink dry: 1.7tbps ddos attack makes history," 2018. [Online]. Available: https://www.netscout.com/blog/security-17tbps-ddos-attack-makes-history

[17] "Aws shield: Threat landscape report," 2020. [Online]. Available: https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

[18] J. Arteaga, "Cldap reflection ddos." [Online]. Available: https://www.akamai.com/uk/en/resources/our-thinking/threat-advisories/connection-less-lightweight-directory-access-protocol-reflection-ddos-threat-advisory.jsp

[19] P. Nicholson, "Five most famous ddos attacks and then some," 2021. [Online]. Available: https://www.a10networks.com/blog/5-most-famous-ddos-attacks

[20] M. Kan, "Microsoft mitigates 3.47tbps ddos attack, a new record," 2022. [Online]. Available: https://www.pcmag.com/news/microsoft-mitigates-347tbps-ddos-attack-a-new-record

[21] Y. Li, Y. Wang, F. Yang, and S. Su, "Traceback drdos attacks," *Journal of Information and Computational Science*, vol. 8, pp. 94–111, 01 2011.

[22] C. Rossow, "Amplification hell: Revisiting network protocols for ddos abuse," 01 2014.

[23] "Github blog, ddos incident report," 2018. [Online]. Available: https://github.blog/2018-03-01-ddos-incident-report

[24] "Defending against carpet bombing ddos attacks." [Online]. Available: https://www.netscout.com/use-case/carpet-bombing-attacks

[25] S. Bjarnason, "Ddos defences in the terabit era: Attack trends, carpet bombing," 2018. [Online]. Available: https://blog.apnic.net/2018/12/04/ddos-defences-in-the-terabit-era-attack-trends-carpet-bombing/

[26] Z. Duan, X. Yuan, and J. Chandrashekar, "Constructing inter-domain packet filters to control ip spoofing based on bgp updates," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 2006, pp. 1–12.

[27] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 15–26. [Online]. Available: https://doi.org/10.1145/383059.383061

[28] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed ip traffic using hop-count filtering," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 40–53, 2007.

[29] F. Baker and P. Savola, "Rfc3704: Ingress filtering for multihomed networks," USA, 2004.

[30] A. Bremler-barr and H. Levy, "Spoofing prevention method," vol. 1, 04 2005, pp. 536 – 547 vol. 1.

[31] X. Liu, X. Yang, D. Wetherall, and T. Anderson, "Efficient and secure source authentication with packet passports," in *2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI 06)*. San Jose, CA: USENIX Association, Jul. 2006. [Online]. Available: https://www.usenix.org/conference/sruti-06/efficient-and-secure-source-authentication-packet-passports

[32] A. Yaar, A. Perrig, and D. Song, "Stackpi: New packet marking and filtering mechanisms for ddos and ip spoofing defense," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1853–1863, 2006.

[33] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "Save: source address validity enforcement protocol," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2002, pp. 1557–1566.

[34] J. Kwon, D. Seo, M. Kwon, H. Lee, A. Perrig, and H. Kim, "An incrementally deployable anti-spoofing mechanism for software-defined networks," *Computer Communications*, vol. 64, 04 2015.

[35] G. Yao, J. Bi, and P. Xiao, "Vase: Filtering ip spoofing traffic with agility," *Computer Networks*, vol. 57, p. 243–257, 01 2013.

[36] K. Sriram, D. Montgomery, and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding," RFC 8704, Feb. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8704

[37] P. R. Marques, J. Mauch, N. Sheth, B. Greene, R. Raszuk, and D. R. McPherson, "Dissemination of Flow Specification Rules," RFC 5575, Aug. 2009. [Online]. Available: https://www.rfc-editor.org/info/rfc5575

[38] W. Kumari, "Rfc 5635-remote triggered black hole filtering with urpf," 2009.

[39] "Cisco. remotely triggered black hole filtering - destination based and source based," 2005. [Online]. Available: http://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf

[40] D. Wagner, D. Kopp, M. Wichtlhuber, C. Dietzel, O. Hohlfeld, G. Smaragdakis, and A. Feldmann, "United we stand: Collaborative detection and mitigation of amplification ddos attacks at scale," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 970–987. [Online]. Available: https://doi.org/10.1145/3460120.3485385

[41] C. Dietzel, M. Wichtlhuber, G. Smaragdakis, and A. Feldmann, "Stellar: Network attack mitigation using advanced blackholing," in *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 152–164. [Online]. Available: https://doi.org/10.1145/3281411.3281413

[42] "Ddos mitigation services." [Online]. Available: https://www.gartner.com/reviews/market/ddos-mitigation-services

[43] "Radware locations," 2022. [Online]. Available: https://www.radware.com/locations/?utm_source=getstartedpageutm_campaign=getstarttab-ddos-protection

[44] "Industry leading ddos protection," 2020. [Online]. Available: https://www.cloudflare.com/cloudflare-ddos-2020/

[45] T. Reddy.K, M. Boucadair, and J. Shallow, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Call Home," RFC 9066, Dec. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9066

[46] "kentik ddos detection." [Online]. Available: https://www.kentik.com/kentipedia/ddos-detection/

[47] "Fastnetmon ddos detection." [Online]. Available: https://fastnetmon.com/

[48] "Radware ddos protection." [Online]. Available: https://www.radware.com/solutions/ddos-protection/

[49] "Solarwinds." [Online]. Available: https://www.solarwinds.com/security-event-manager/use-cases/ddos-attack

[50] "Bgp in 2021 – bgp updates," 2021. [Online]. Available: https://labs.apnic.net/?p=1559

[51] "Caida as relationship data," 2019. [Online]. Available: https://publicdata.caida.org/datasets/as-relationships/serial-2/20190801.as-rel2.txt.bz2

[52] M. Sabag, "Bgp simulator." [Online]. Available: https://github.com/MatanSabag/BgpSimulator

[53] Y. Gilad, "disco," 2020. [Online]. Available: https://github.com/yossigi/disco

[54] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733–745, 2001.

[55] "Shodan: Search engine for the internet of everything." [Online]. Available: https://www.shodan.io/

[56] M. Boucadair and T. Reddy.K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification," RFC 8783, May 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8783

[57] R. Canonico, D. Cotroneo, L. Peluso, S. Romano, and G. Ventre, "Programming routers to improve network security," in *Proceedings of the OPENSIG 2001 Workshop Next Generation Network Programming*. Citeseer, 2001.

[58] S. Chen and Q. Song, "Perimeter-based defense against high bandwidth ddos attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 6, pp. 526–537, 2005.

[59] J. Freudiger, E. De Cristofaro, and A. Brito, "Controlled data sharing for collaborative predictive blacklisting," 02 2015.

[60] S. Katti, B. Krishnamurthy, and D. Katabi, "Collaborating against common enemies," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. USA: USENIX Association, 2005, p. 34.

[61] L. Melis, G. Danezis, and E. D. Cristofaro, "Efficient private statistics with succinct sketches," *CoRR*, vol. abs/1508.06110, 2015. [Online]. Available: http://arxiv.org/abs/1508.06110